EDITS: Modeling and Mitigating Data Bias for Graph Neural Networks

Yushun Dong
University of Virginia
Charlottesville, USA
yd6eb@virginia.edu

Ninghao Liu
University of Georgia
Athens, USA
ninghao.liu@uga.edu

Brian Jalaian
Army Research Laboratory
Arlington, USA
brian.a.jalaian.civ@mail.mil

Jundong Li
University of Virginia
Charlottesville, USA
jundong@virginia.edu

ABSTRACT

Graph Neural Networks (GNNs) have recently demonstrated superior capability of tackling graph analytical problems in various applications. Nevertheless, with the wide-spreading practice of GNNs in high-stake decision-making processes, there is an increasing societal concern that GNNs could make discriminatory decisions that may be illegal towards certain demographic groups. Although some explorations have been made towards developing fair GNNs, existing approaches are tailored for a specific GNN model. However, in practical scenarios, myriads of GNN variants have been proposed for different tasks, and it is costly to train and fine-tune existing debiasing models for different GNNs. Also, bias in a trained model could originate from training data, while how to mitigate bias in the graph data is usually overlooked. In this work, different from existing work, we first propose novel definitions and metrics to measure the bias in an attributed network, which leads to the optimization objective to mitigate bias. Based on the optimization objective, we develop a framework named EDITS to mitigate the bias in attributed networks while preserving useful information. EDITS works in a model-agnostic manner, which means that it is independent of the specific GNNs applied for downstream tasks. Extensive experiments on both synthetic and real-world datasets demonstrate the validity of the proposed bias metrics and the superiority of EDITS on both bias mitigation and utility maintenance. Open-source implementation: https://github.com/yushundong/EDITS.

1 INTRODUCTION

Graph-structured data, such as social networks [34, 52], chemical reaction networks [13] and traffic networks [49], has become ubiquitous in a plethora of realms. To better understand graph data, various graph mining models have been proposed. Among them, the recently emerged Graph Neural Networks (GNNs) have demonstrated superior capability of tackling graph analytical problems in various tasks, such as node classification [29, 54, 55] and link prediction [30, 62]. Despite the superior performance, GNNs usually do not consider fairness issues in the learning process [12]. Extensive research has been carried out to demonstrate that graph mining algorithms, especially those recently proposed GNNs [12, 51, 59], could deliver discriminatory decisions dominated by sensitive attributes such as gender [17] and political preference [43] in many decision-making systems. In real-world scenarios, people are unwilling to experience such discriminatory decisions, which sometimes are even in conflict with anti-discrimination laws [26]. As a consequence, the wide-spreading practice of GNNs has turned it into a crucial problem to mitigate bias in real applications.

Various efforts have been made to mitigate the bias exhibited in graph mining algorithms. For example, random walk algorithm can be modified via improving the appearance rate of minorities [8, 48]. Adversarial learning is another popular approach, which aims to learn node embeddings that are not distinguishable on sensitive attributes [6, 40]. Some recent efforts have also been made to mitigate bias in the outcome of GNNs. For example, adversarial learning can also be adapted to GNNs for outcome bias mitigation [12]. Nevertheless, existing approaches to debias GNN outcome are tailored for a specific GNN model on a certain downstream task. In practical scenarios, different tasks could adopt different GNN variants, and it is costly to train and fine-tune the debiasing models for diverse GNN backbones. As a consequence, to mitigate bias more efficiently for different GNNs and tasks, developing an one-size-fits-all approach becomes highly desired. Then the question is: how can we perform debiasing regardless of specific GNNs and downstream tasks? Considering that a model trained on biased datasets also tends to be biased [5, 12, 15], directly debiasing the dataset itself can be a straightforward solution. There are already debiasing approaches modifying original datasets via perturbing data distributions or reweighting the data points in the dataset [9, 26, 57]. These approaches obtain less biased datasets, which help to mitigate bias in downstream tasks. In this regard, considering that debiasing for different GNNs is costly, it is also highly desired to mitigate the bias in attributed networks before they are fed into GNNs. Nevertheless, to the best of our knowledge, despite its fundamental importance, no existing literature has taken such a step forward.

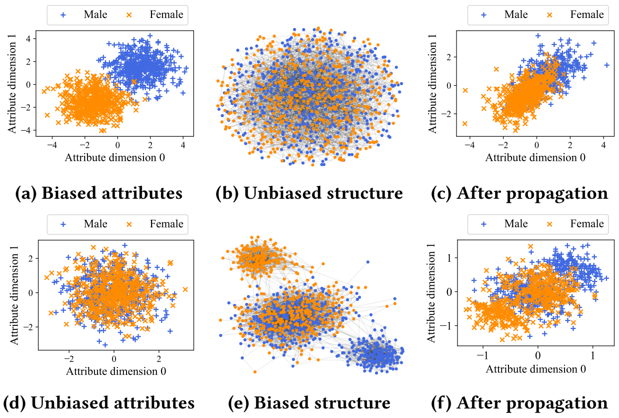
In this paper, we aim to make an initial investigation on debiasing attributed networks as the input of GNNs. Specifically, we tackle the following challenges in debiasing attributed networks. (1) ***Data Bias Modeling***. Traditionally, bias modeling is coupled with the outcomes of a specific downstream task, e.g., node classification. Based on such outcomes, bias can then be modeled via metrics focusing on different notions, e.g., *Statistical Parity* [15] and *Equality of Opportunity* [21], to determine whether the outcome is biased towards some specific demographic groups. Nevertheless, if debiasing is carried out directly based on the attributed network data, the first and foremost challenge is how to appropriately model such data bias. (2) ***Multi-modality Debiasing***. In fact, attributed networks contain both graph structure and node attributes information. Correspondingly, bias may exist with diverse formats across different data modalities. In this regard, how to debias attributed networks with the existence of different data modalities is the second challenge that needs to be tackled. (3) ***Model-agnostic Debiasing***. Existing debiasing approaches for GNNs commonly require supervision signal in the training process. Different from these approaches, model-agnostic debiasing for GNNs does not require the supervision over model output, as they are decoupled from each other. Nevertheless, the ultimate goal of debiasing still lies in mitigating the bias in model output of a specific downstream task. Such a contradiction poses the challenge of how to properly formulate a debiasing objective that can be universally applied to different GNNs in downstream tasks.

To tackle the challenges discussed above, we present novel data bias modeling approaches and a principled debiasing framework named EDITS (modEling anD mItigating daTa biaS) to achieve model-agnostic attributed network debiasing for GNNs. Specifically, we first carry out preliminary analysis to illustrate how bias exists in the two data modalities (i.e., node attributes and network structure) and affects each other in the information propagation of GNNs. Then, we formally define *attribute bias* and *structural bias*, together with the corresponding metrics for data bias modeling. Besides, we formulate the problem of debiasing attributed networks for GNNs, and propose a novel framework named EDITS for bias mitigation. It should be noted that EDITS is model-agnostic for different GNNs, meaning that we directly mitigate bias for attributed networks. Finally, empirical evaluations on both synthetic and real-world datasets corroborate the validity of the proposed bias metrics and the effectiveness of EDITS. Our contributions are summarized as:

- **Problem Formulation.** We formulate and make an initial investigation on a novel problem: debiasing attributed networks for GNNs based on the analysis of the information propagation mechanism of GNNs.
- **Metric and Algorithm Design.** We design new metrics to model the bias existed in both data modalities of attributed networks, and propose a debiasing framework named EDITS to mitigate bias in attributed networks for GNNs.
- **Theoretical Analysis.** We provide theoretical analysis for the proposed metrics and framework, which not only builds connections between the two kinds of bias, but also reveal the essence of the proposed framework EDITS.
- **Experimental Evaluation.** We conduct comprehensive experiments on both synthetic and real-world datasets to verify the validity of the proposed bias metrics and the effectiveness of the proposed framework EDITS.

2 PRELIMINARY ANALYSIS

To show how the two data modalities (i.e., attribute and structure) introduce bias together in an attributed network during information propagation – the most common operation in GNNs, here we provide two exemplary cases. These two cases also bring insights on tackling the three challenges mentioned in Section 1. Following previous literature on algorithmic bias [15], the attribute distribution difference between groups is regarded as the bias in attribute, while the group membership distribution difference of the neighbors for nodes between groups is regarded as the bias in structure. Such bias in attribute and structure can be regarded as the bias existed in two data modalities in an attributed network. To better illustrate how biased and unbiased data modalities affect each other, we only introduce bias to one data modality in each case, and leave the other data modality unbiased. Now we explain how synthetic attributed networks are generated. Without loss of generality, we assume the *sensitive attribute* is gender, and 1,000 nodes are generated with half males (blue) and half females (orange) for both cases. In addition to the sensitive attribute, each node is with an extra two-dimensional attribute vector, which will be initialized and utilized for information propagation. To introduce bias to either of the data modalities, different strategies are adopted to generate the attribute vector and the network structure in the two cases. To study how biased and unbiased data modalities affect each other during information propagation, we compare the distribution difference of attribute between groups before and after propagation. Here the information propagation mechanism of GCN [29] is adopted for both cases.



(a) Biased attributes **(b) Unbiased structure** **(c) After propagation**

(d) Unbiased attributes **(e) Biased structure** **(f) After propagation**

Figure 1: Two exemplary cases illustrating how bias in two data modalities affect each other in GNN information propagation. Here (c) is the attribute distribution after propagation with biased attributes (a) and unbiased structure (b); while (f) is the attribute distribution after propagation with unbiased attributes (d) and biased structure (e).

Case 1: Biased attributes and unbiased structure. In this case, we generate biased two-dimensional attribute vectors for nodes from the two groups (i.e., males and females) and unbiased network structure. Specifically, biased attributes at each dimension is generated independently with Gaussian distribution $\mathcal{N}(-1.5,\ 1^2)$ for female and $\mathcal{N}(1.5,\ 1^2)$ for male. The distributions are shown in Fig. (1a). We then introduce how unbiased network structure is generated. For each node in an unbiased network structure, the expected distribution of the group membership ratio in its neighbor set should be independent with the group membership of the node itself. In this regard, we generate unbiased network structure via *random graph* model with edge formation probability as 2×10^{-3}. The visualization of the network is presented in Fig. (1b). The attribute distribution after information propagation w.r.t. the network structure is shown in Fig. (1c). Comparing Fig. (1a) (attribute distribution before propagation) with (1c) (attribute distribution after propagation), we can see the unbiased structure helps mitigate the original attribute bias after attributes are propagated following network topology. This not only implies that the attribute distribution difference between groups is a vital source of bias, but also demonstrates that unbiased structure helps mitigate bias in attributes after the information propagation process.

Case 2: Unbiased attributes and biased structure. In this case, unbiased attributes are generated independently at each dimension with $\mathcal{N}(0, 1^2)$ for both males and females. The distributions are shown in Fig. (1d). Biased network structure is generated as follows. For each node, we sum up its attribute values. Then, we rank all nodes in a descending order according to the attribute sum. After that, given a threshold integer t, for the top-ranked t males and bottom-ranked t females, we assume that they form two separated communities. The two communities are shown as the bottom right community (males) and the upper left community (females) in Fig. (1e). We generate edges via *random graph* model with edge formation probability as 5×10^{-2} within each community. Similarly, the rest nodes form the third community via *random graph* model with edge formation probability as 1×10^{-2}. We also generate edges between nodes from the male (or female) community and the third community with the probability of 2×10^{-4}. In this way, we introduce bias in network structure. The final network is presented in Fig. (1e). The attribute distribution after propagation w.r.t. the network structure is shown in Fig. (1f). Comparing Fig. (1d) with (1f), we find that even if the original attributes are unbiased, the biased structure still turns the attributes into biased ones after information propagation. This implies that the bias contained in the network structure is also a significant source of bias.

Based on the above discussions, we draw three preliminary conclusions to help us tackle the challenges mentioned in Section 1. (1) For ***Data Bias Modeling***, bias in attributes can be modeled based on the difference of attribute distribution between groups. Also, bias in network structure can be modeled based on the difference of attribute distribution between groups after information propagation. (2) For ***Multi-modality Debiasing*** in an attributed network, at least two debiasing processes should be carried out targeting the two data modalities (i.e., attributes and structure). (3) For ***Model-agnostic Debiasing***, if the attribute distributions between groups can be less biased both before and after general information propagation, the learned node representations tend to be indistinguishable between groups. Then, GNNs trained on such data could also be less biased.

3 MODELING DATA BIAS FOR GNNS

In this section, consistent with traditional algorithmic bias notion [15], we define *attribute bias* and *structural bias* in attributed networks together with their metrics. Without loss of generality, only binary sensitive attributes are considered, which can be easily generalized to more complicated scenarios. Theoretical analysis corresponding to the metrics will be presented in Section 5.

3.1 Preliminaries

In this paper, without further specification, bold uppercase letters (e.g., $\mathbf{X}$), bold lowercase letters (e.g., $\mathbf{x}$), and normal lowercase letters (e.g., x) represent matrices, vectors, and scalars, respectively. For any matrix, e.g., $\mathbf{X}$, we use $\mathbf{X}_i$ denote its i-th row.

Let $\mathcal{G} = (\mathbf{A}, \mathbf{X})$ be an undirected attributed network. $\mathbf{A} \in \mathbb{R}^{N \times N}$ is the adjacency matrix, and $\mathbf{X} \in \mathbb{R}^{N \times M}$ is the node attribute matrix, where N is the number of nodes and M is the attribute dimension. Let diagonal matrix $\mathbf{D}$ be the degree matrix of $\mathbf{A}$, where its (i,i)-th entry $\mathbf{D}_{i,i} = \sum_j \mathbf{A}_{ij}$, and $\mathbf{D}_{i,j} = 0$ $(i \neq j)$. $\mathbf{L} = \mathbf{D} - \mathbf{A}$ is the graph Laplacian. Denote the normalized adjacency matrix and the normalize Laplacian matrix as $\mathbf{A}_{\text{norm}} = \mathbf{D}^{-\frac{1}{2}}\mathbf{A}\mathbf{D}^{-\frac{1}{2}}$ and $\mathbf{L}_{\text{norm}} = \mathbf{D}^{-\frac{1}{2}}\mathbf{L}\mathbf{D}^{-\frac{1}{2}}$. $|.|$ is the absolute value operator for scalars.

3.2 Definitions of Bias

We consider two types of bias on attributed networks, i.e., attribute bias and structural bias. We first define attribute bias as follows.

Definition 1. ***Attribute bias.*** *Given an undirected attributed network $\mathcal{G} = (\mathbf{A}, \mathbf{X})$ and the corresponding group indicator (w.r.t. the sensitive attribute) for each node $\mathbf{s} = [s_1, s_2, ..., s_N]$, where $s_i \in \{0, 1\}$ $(1 \leq i \leq N)$. For any attribute, if the distributions between different demographic groups are different, then attribute bias exists in $\mathcal{G}$.*

Besides, as shown in the second example in Section 2, bias can also emerge after attributes are propagated in the network even when original attributes are unbiased. Therefore, an intuitive idea to identify structural bias is whether information propagation in the network introduces or exacerbates bias [24]. Formally, we give the definition of structural bias on attributed networks as follows.

Definition 2. ***Structural bias.*** *Given an undirected attributed network $\mathcal{G} = (\mathbf{A}, \mathbf{X})$ and the corresponding group indicator (w.r.t. sensitive attribute) for each node $\mathbf{s} = [s_1, s_2, ..., s_N]$, where $s_i \in \{0, 1\}$ $(1 \leq i \leq N)$. If any information propagation w.r.t. $\mathbf{A}$ promotes the distribution difference between different groups at any attribute dimension, then structural bias exists in $\mathcal{G}$.*

Apart from these definitions, it is also necessary to quantitatively measure the attribute bias and structural bias. In the sequel, we introduce our proposed metrics for the two types of bias.

3.3 Bias Metrics

Here we take the first step to define metrics for both *attribute bias* and *structural bias* for an undirected attributed network $\mathcal{G}$. We defer a detailed theoretical analysis that builds the connections between these two bias metrics in Section 5.

Attribute bias metric. Let $\mathbf{X}_{\text{norm}} \in \mathbb{R}^{N \times M}$ be the column normalized attribute matrix. For the m-th dimension $(1 \leq m \leq M)$ of $\mathbf{X}_{\text{norm}}$, we use $\mathcal{X}_m^0$ and $\mathcal{X}_m^1$ to denote attribute value set for nodes with $s_i = 0$ and $s_i = 1$ $(1 \leq i \leq N)$, respectively. Then we have $\mathcal{X}_{total} = \{(\mathcal{X}_1^0, \mathcal{X}_1^1), (\mathcal{X}_2^0, \mathcal{X}_2^1), ..., (\mathcal{X}_M^0, \mathcal{X}_M^1)\}$ for node attributes from two groups. We measure attribute bias with Wasserstein distance between the distributions of the two groups:

$$b_{\text{attr}} = \frac{1}{M} \sum_m W(pdf(\mathcal{X}_m^0), pdf(\mathcal{X}_m^1)). \tag{1}$$

Here $pdf(\cdot)$ is the probability density function for a set of values, and $W(.,.)$ is the Wasserstein distance between two distributions. Intuitively, b_{attr} describes the average Wasserstein distance between attribute distributions of different groups across all dimensions, which is consistent with traditional algorithmic bias notion [15].

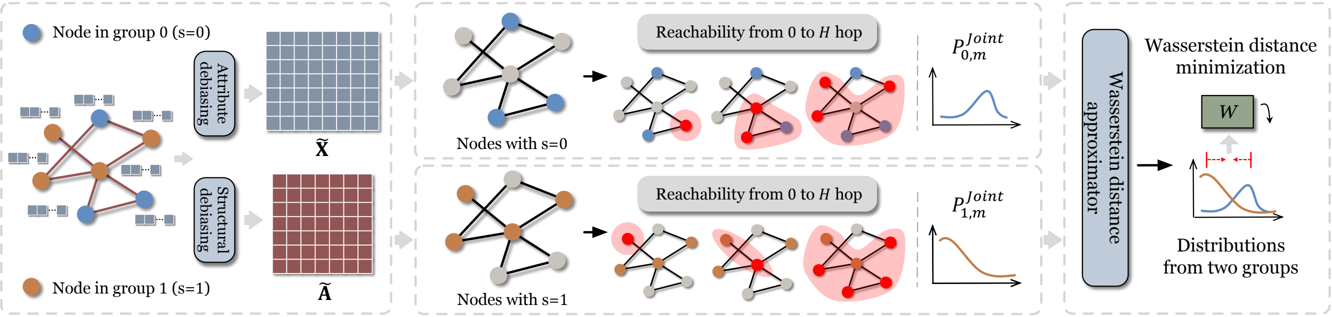

Node in group 0 (s=0)
Node in group 1 (s=1)
Attribute debiasing
Structural debiasing
X̃
Ã
Nodes with s=0
Nodes with s=1
Reachability from 0 to H hop
Reachability from 0 to H hop
$P_{0,m}^{Joint}$
$P_{1,m}^{Joint}$
Wasserstein distance approximator
Wasserstein distance minimization
W
Distributions from two groups

Figure 2: An illustration of the proposed framework EDITS with $H = 2$ as an example.

Structural bias metric. As illustrated in Section 2, the key mechanism of GNNs is information propagation, where structural bias could exhibit. Let $\mathrm{P}_{\text{norm}} = \alpha\mathbf{A}_{\text{norm}} + (1-\alpha)\mathbf{I}$. Here P_{norm} can be regarded as a normalized adjacency matrix with re-weighted self-loops, where $\alpha \in [0,1]$ is a hyper-parameter. Before measuring structural bias, we define the *propagation matrix* $\mathrm{M}_H \in \mathbb{R}^{N\times N}$ as:

$$\mathrm{M}_H = \beta_1 \mathrm{P}_{\text{norm}} + \beta_2 \mathrm{P}_{\text{norm}}^2 + \ldots + \beta_H \mathrm{P}_{\text{norm}}^H, \tag{2}$$

where β_h $(1 \le h \le H)$ is the re-weighting parameter. The rationale behind the formulation above is to measure the aggregated reaching likelihood from each node to other nodes within a distance of H. To achieve localized effect for each node, a desired choice is to let $\beta_1 \ge \beta_2 \ge \ldots \ge \beta_H$, i.e., emphasizing small-hop terms and reducing the weights of large-hop terms. For example, assume $H = 3$, so the value $(\mathrm{M}_3)_{i,j}$ is the aggregated reaching likelihood from node i to node j within 3 hops with re-weighting parameters being β_1, β_2 and β_3. Also, given attributes X_{norm}, we define the *reachability matrix* $\mathrm{R} \in \mathbb{R}^{N\times M}$ as $\mathrm{R} = \mathrm{M}_H \mathrm{X}_{\text{norm}}$. Intuitively, $\mathrm{R}_{i,m}$ is the aggregated reachable attribute value for attribute m of node i. Similar to attribute bias, we utilize $\mathcal{R}_m^0$ and $\mathcal{R}_m^1$ to represent the set of values of the m-th dimension in R for nodes with $s_i = 0$ and $s_i = 1$ $(1 \le i \le N)$, respectively. Then, the entries in R can also be divided into tuples according to attribute dimensions: $\mathcal{R}_{total} = \{(\mathcal{R}_1^0, \mathcal{R}_1^1), (\mathcal{R}_2^0, \mathcal{R}_2^1), ..., (\mathcal{R}_M^0, \mathcal{R}_M^1)\}$. We define structural bias as:

$$b_{\text{stru}} = \frac{1}{M}\sum_m W(pdf(\mathcal{R}_m^0), pdf(\mathcal{R}_m^1)). \tag{3}$$

Here b_{stru} is defined in a similar way as b_{attr}, except that the former uses $\mathcal{R}_m^0$ and $\mathcal{R}_m^1$ instead of $\mathcal{X}_m^0$ and $\mathcal{X}_m^1$. In this way, structural bias b_{stru} describes the average difference between aggregated attribute distributions of different groups after several rounds of propagation, which also agrees with traditional algorithmic bias notion [15].

3.4 Problem Statement

Based on the definitions and metrics in Section 3.2 and Section 3.3, we argue that if both b_{attr} and b_{stru} are reduced, bias in an attributed network can be mitigated. As a result, if GNNs are trained on such data, the bias issues in downstream tasks could also be alleviated. Formally, we define the debiasing problem as follows.

PROBLEM 1. ***Debiasing attributed networks for GNNs.*** *Given an attributed network $\mathcal{G} = (\mathrm{A}, \mathrm{X})$, our goal is to debias $\mathcal{G}$ by reducing b_{attr} and b_{stru} to obtain $\tilde{\mathcal{G}} = (\tilde{\mathrm{A}}, \tilde{\mathrm{X}})$, so that the bias of GNNs trained on $\tilde{\mathcal{G}}$ is also mitigated. The debiasing process is indepdent of detailed architecture and parameterization of GNNs.*

4 MITIGATING DATA BIAS FOR GNNS

In this section, we discuss how to tackle Problem 1 with our proposed framework EDITS. We first present an overview of EDITS, followed by the formulation of the objective function. Finally, we present the optimization process for the objective.

4.1 Framework Overview

An overview of the proposed framework EDITS is shown in Fig. (2). Specifically, EDITS consists of three learnable modules. Gradient descent based optimization is executed alternatively in training.

- ***Attribute debiasing.*** This module learns a debiasing function $g_{\boldsymbol{\theta}}$ with learnable parameter $\boldsymbol{\theta} \in \mathbb{R}^M$. The debiased version of X is obtained as output where $\tilde{\mathrm{X}} = g_{\boldsymbol{\theta}}(\mathrm{X})$.
- ***Structural debiasing.*** This module outputs $\tilde{\mathrm{A}}$ as the debiased A. Specifically, $\tilde{\mathrm{A}}$ is initialized with A at the beginning of the optimization process. The entries in $\tilde{\mathrm{A}}$ are optimized via gradient descent with clipping and binarization.
- ***Wasserstein distance approximator.*** This module learns a function f for each attribute dimension. Here f is utilized to estimate the Wasserstein distance between the distributions of different groups for any attribute dimension.

4.2 Objective Function

In this subsection, we introduce the details of our framework. Following the Definition 1 and Definition 2, our goal is to reduce b_{attr} and b_{stru} simultaneously. For the ease of understanding, we first only consider the m-th attribute dimension as an example, and then extend it to all M dimensions to obtain our final objective.

Let $x_{0,m} \sim P_{0,m}$ and $x_{1,m} \sim P_{1,m}$ be the random variables that represent the value of the m-th attribute dimension in X for nodes with sensitive attribute $s = 0$ and $s = 1$, respectively. Assume that we have a function $g_{\theta_m} : \mathbb{R} \to \mathbb{R}$ $(1 \le m \le M)$ to mitigate attribute bias. For the m-th dimension, we denote $x_{0,m}^{(0)} = g_{\theta_m}(x_{0,m}) \sim P_{0,m}^{(0)}$ and $x_{1,m}^{(0)} = g_{\theta_m}(x_{1,m}) \sim P_{1,m}^{(0)}$ as the debiasing results for $x_{0,m}$ and $x_{1,m}$, respectively. Here the superscript (0) indicates that no information propagation is performed in the debaising process. Correspondingly, when such operation is extended to all M dimensions, we will have the debiased attribute matrix $\tilde{\mathrm{X}}$. Apart from the goal of mitigating attribute bias, we also want to mitigate structural

Algorithm 1 The Optimization Algorithm for EDITS

```
Input:
    A: Adjacency matrix; X: Attribute matrix; $\alpha$, $\mu_1$ to $\mu_4$: Hyper-parameters in objectives; $c$: Threshold enforcing lipschitz; $z$: Threshold for attribute masking; $r$: Threshold factor for adjacency matrix binarizing;
Output:
    Debiased adjacency matrix $\tilde{\mathrm{A}}$ and attribute matrix $\tilde{\mathrm{X}}$;
$\tilde{\mathrm{A}} \leftarrow \mathrm{A}$; $\mathbf{\Theta} \leftarrow \mathrm{I}$;
while $epoch \leq epoch\_max$ do
    Compute $\mathscr{L}_1$ following Eq. (9);
    Update the weights of $f_m$ by SGD following Eq. (10);
    Clip the weights of $f$ within [-$c$, $c$];
    Update $\mathbf{\Theta}$ by PGD following Eq. (11), $\tilde{\mathrm{X}} \leftarrow \mathrm{X}\mathbf{\Theta}$;
    Update $\tilde{\mathrm{A}}$ by PGD following Eq. (12), $\tilde{\mathrm{A}} \leftarrow \frac{1}{2}(\tilde{\mathrm{A}} + \tilde{\mathrm{A}}^{\top})$;
end while
Mask the $z$ smallest entries with 0 in $diag(\mathbf{\Theta})$, $\tilde{\mathrm{X}} \leftarrow \mathrm{X}\mathbf{\Theta}$;
Binarize $\tilde{\mathrm{A}}$ w.r.t. the threshold $r$;
return $\tilde{\mathrm{A}}$ and $\tilde{\mathrm{X}}$;
```

bias. Let $\tilde{\mathrm{A}}$ be the adjacency matrix from the debiased network structure, and $\tilde{\mathrm{P}}_{\mathrm{norm}}$ denotes the normalized $\tilde{\mathrm{A}}$ with re-weighted self-loops. Information propagation with h hops using the debiased adjacency matrix could be expressed as $\tilde{\mathrm{P}}^{h}_{\mathrm{norm}}\tilde{\mathrm{X}}$, where $1 \leq h \leq H$. Let $x^{(h)}_{0,m} \sim P^{(h)}_{0,m}$ and $x^{(h)}_{1,m} \sim P^{(h)}_{1,m}$ be the random variables of the value at the m-th column of $\tilde{\mathrm{P}}^{h}_{\mathrm{norm}}\tilde{\mathrm{X}}$ for nodes with sensitive attribute $s = 0$ and $s = 1$, respectively. We hope that $\tilde{\mathrm{A}}$ could mitigate structural bias. In this work, we combine attribute debiasing and structure debiasing as below.

Based on the random variables $x^{(0)}_{0,m}$ to $x^{(H)}_{0,m}$ and $x^{(0)}_{1,m}$ to $x^{(H)}_{1,m}$, we have the $(H+1)$-dimensional vectors $\mathrm{x}_{0,m} = [x^{(0)}_{0,m}, x^{(1)}_{0,m}, ..., x^{(H)}_{0,m}]$ and $\mathrm{x}_{1,m} = [x^{(0)}_{1,m}, x^{(1)}_{1,m}, ..., x^{(H)}_{1,m}]$ following the joint distribution $P^{Joint}_{0,m}$ and $P^{Joint}_{1,m}$, respectively. To reduce both b_{attr} and b_{stru} at the m-th dimension, our goal is to minimize the Wasserstein distance between $P^{Joint}_{0,m}$ and $P^{Joint}_{1,m}$:

$$\min_{\theta_m, \tilde{\mathrm{A}}} W(P^{Joint}_{0,m}, P^{Joint}_{1,m}), \tag{4}$$

where $W(P^{Joint}_{0,m}, P^{Joint}_{1,m})$ can be expressed as

$$W(P^{Joint}_{0,m}, P^{Joint}_{1,m}) = \inf_{\gamma \in \Pi(P^{Joint}_{0,m}, P^{Joint}_{1,m})} \mathbb{E}_{(\mathrm{x}_{0,m}, \mathrm{x}_{1,m}) \sim \gamma}[\|\mathrm{x}_{0,m} - \mathrm{x}_{1,m}\|_1]. \tag{5}$$

Here $\Pi(P^{Joint}_{0,m}, P^{Joint}_{1,m})$ represents the set of all joint distributions $\gamma(\mathrm{x}_{0,m}, \mathrm{x}_{1,m})$ whose marginals are $P^{Joint}_{0,m}$ and $P^{Joint}_{1,m}$, respectively. After considering all the M dimensions, the overall objective is

$$\min_{\boldsymbol{\theta}, \tilde{\mathrm{A}}} \frac{1}{M} \sum_{1 \leq m \leq M} W(P^{Joint}_{0,m}, P^{Joint}_{1,m}). \tag{6}$$

It is non-trivial to optimize Eq. (6) as the infimum is intractable. Therefore, in the next subsection, we show how to convert it into a tractable optimization problem through approximation, which enables end-to-end gradient-based optimization.

4.3 Framework Optimization

In this subsection, we introduce our optimization algorithm. For simplicity, first we still use the m-th attribute dimension in X to illustrate the idea. Considering the infimum in Wasserstein distance computation is intractable, we apply the Kantorovich-Rubinstein duality [56] to convert the problem of Eq. (5) as:

$$W(P^{Joint}_{0,m}, P^{Joint}_{1,m}) = \sup_{\|f\|_L \leq 1} \mathbb{E}_{\mathrm{x}_{0,m} \sim P^{Joint}_{0,m}}[f(\mathrm{x}_{0,m})] - \mathbb{E}_{\mathrm{x}_{1,m} \sim P^{Joint}_{1,m}}[f(\mathrm{x}_{1,m})]. \tag{7}$$

Here $\|f\|_L \leq 1$ denotes that the supremum is taken over all 1-Lipschitz functions $f: \mathbb{R}^{H+1} \longrightarrow \mathbb{R}$. The problem can be solved by learning a neural network as f. Nevertheless, it is worth noting that the 1-Lipschitz function is difficult to obtain during optimization. Therefore, here we relax $\|f\|_L \leq 1$ to $\|f\|_L \leq k$ (k is a constant). In this case, the left side of Eq. (7) also changes to $kW(P^{Joint}_{0,m}, P^{Joint}_{1,m})$. Then, the Wasserstein distance between $P^{Joint}_{0,m}$ and $P^{Joint}_{1,m}$ up to a multiplicative constant can be attained via:

$$\max_{f_m \in \mathcal{F}} \mathbb{E}_{\mathrm{x}_{0,m} \sim P^{Joint}_{0,m}}[f_m(\mathrm{x}_{0,m})] - \mathbb{E}_{\mathrm{x}_{1,m} \sim P^{Joint}_{1,m}}[f_m(\mathrm{x}_{1,m})], \tag{8}$$

where $\mathcal{F}$ denotes the set of all k-Lipschitz functions (i.e., $\|f_m\|_L \leq k$, $f_m \in \mathcal{F}$). Then, extending Eq. (8) to all M dimensions leads to our final objective function as:

$$\mathscr{L}_1 = \sum_{1 \leq m \leq M} \{\mathbb{E}_{\mathrm{x}_{0,m} \sim P^{Joint}_{0,m}}[f_m(\mathrm{x}_{0,m})] - \mathbb{E}_{\mathrm{x}_{1,m} \sim P^{Joint}_{1,m}}[f_m(\mathrm{x}_{1,m})]\}, \tag{9}$$

where $\{f_m : 1 \leq m \leq M\} \subset \mathcal{F}$. To model the function f in Eq. (9), we use a single-layered neural network for the approximation of each f_m ($1 \leq m \leq M$). Therefore, the goal of *Wasserstein distance approximators* can be formulated as:

$$\max_{\{f_m : 1 \leq m \leq M\} \subset \mathcal{F}} \mathscr{L}_1 \ . \tag{10}$$

The weights of neural networks are clipped within $[-c, c]$ (c is a pre-defined constant), which has been proved to be a simple but effective way to enforce the Lipschitz constraint for every f_m [3]. For attribute debiasing g_θ, we choose a linear function, i.e., $g_{\theta_m}(x_{s,m}) = \theta_m x_{s,m}$ ($s \in \{0, 1\}$). One advantage is that it acts as the role of feature re-weighting by assigning a feature weight for each attribute, which enables better interpretability for the debiased result. In matrix form, assume $\mathbf{\Theta}$ is a diagonal matrix with the m-th diagonal entry being θ_m, we have $\tilde{\mathrm{X}} = g_{\boldsymbol{\theta}}(\mathrm{X}) = \mathrm{X}\mathbf{\Theta}$. Then the optimization goal for *attribute debiasing* is:

$$\min_{\mathbf{\Theta}} \mathscr{L}_1 + \mu_1 \|\tilde{\mathrm{X}} - \mathrm{X}\|^2_F + \mu_2 \|\mathbf{\Theta}\|_1, \tag{11}$$

where μ_1 and μ_2 are hyper-parameters. The second term ensures the debiased attribute after feature re-weighting is close to the original attributes (i.e., preserve as much information as possible). The third term controls the sparsity of re-weighting parameters. For *structural debiasing*, $\tilde{\mathrm{A}}$ is optimized with the objective as:

$$\min_{\tilde{\mathrm{A}}} \mathscr{L}_1 + \mu_3 \|\tilde{\mathrm{A}} - \mathrm{A}\|^2_F + \mu_4 \|\tilde{\mathrm{A}}\|_1 \quad s.t., \tilde{\mathrm{A}} = \tilde{\mathrm{A}}^{\top}. \tag{12}$$

where μ_3 and μ_4 are hyper-parameters. The second term ensures the debiased result $\tilde{\mathrm{A}}$ is close to the original topology A. The third term enforces the debiased network topology is also sparse, which is aligned with the characteristics of real-world networks. To optimize

f, $\mathbf{\Theta}$ and $\tilde{\mathrm{A}}$, we propose a gradient-based optimization approach for alternatively training as Algorithm 1. Stochastic Gradient Descent (SGD) is utilized for the optimization of Eq. (10), while Proximal Gradient Descent (PGD) is adopted for solving Eq. (11) and Eq. (12), respectively. For the projection operation in PGD, we clip both the parameters in $\mathbf{\Theta}$ and $\tilde{\mathrm{A}}$ within [0, 1]. To remove the most biased attribute channels, the z smallest weights in the diagonal of $\mathbf{\Theta}$ are masked with 0, where z is a pre-assigned hyper-parameter for attribute debiasing. Besides, we binarize the debiased output $\tilde{\mathrm{A}}$ as follows. For the entries in $\tilde{\mathrm{A}}$ that are larger than the corresponding entries in A, we first compute the maximum improvement of the entry pairs in the corresponding positions between $\tilde{\mathrm{A}}$ and A. Then we re-weight such maximum difference by a factor of r, which is a pre-assigned hyper-parameter for structural debiasing. All entries that are improved by the value larger than the re-weighted difference are directly assigned as 1 in $\tilde{\mathrm{A}}$, and those do not exceed the re-weighted difference are assigned as 0. For the entries in $\tilde{\mathrm{A}}$ that are smaller than the corresponding entries in A, we also compute a similar re-weighted maximum decreasing difference, and assign the deceasing entries that exceed this value and those do not as 0 and 1, respectively. To summarize the basic rationale of binarizing $\tilde{\mathrm{A}}$, this operation aims to directly flip (i.e., 0 to 1 or 1 to 0) the entries with the largest changes after optimization compared with A, and maintain other entries as their original values. Finally, Algorithm 1 outputs $\tilde{\mathrm{X}}$ after the dimension with z smallest θ_m being masked, and $\tilde{\mathrm{A}}$ after being binarized w.r.t. the threshold factor r after multiple epochs of optimization.

5 THEORETICAL ANALYSIS

Here we present theoretical analysis to gain a deeper understanding of debiasing attributed networks. Specifically, we focus on bridging the gap between attribute bias and structural bias from the perspective of Spectral Graph Theory [10]. Usually, an undirected attributed network is regarded as a signal composed of different frequency components in Graph Signal Processing (GSP) [47]. An operation preserving lower frequency components more can be regarded as low-pass filtering the input graph signal [41, 47].

Theorem 1. *Let λ_{max} be the largest eigenvalue of L_{norm}. Multiplying X by the propagation matrix M_H can be regarded as low-pass filtering X when $\alpha = \frac{1}{\lambda_{max}}$ and $\beta_i > 0$ $(1 \leq i \leq H)$.*

Proof. Without loss of generality, we present the proof based on Laplacian graph spectrum. By replacing α with $\frac{1}{\lambda_{\max}}$, we have

$$\mathrm{P}_{\text{norm}} = \frac{1}{\lambda_{\max}} \mathrm{A}_{\text{norm}} + (1 - \frac{1}{\lambda_{\max}})\mathrm{I} = \mathrm{I} - \frac{\mathrm{L}_{\text{norm}}}{\lambda_{\max}}. \tag{13}$$

Then, by combining Eq. (2) and Eq. (13), we get

$$\mathrm{M}_H = \beta_1 (\mathrm{I} - \frac{\mathrm{L}_{\text{norm}}}{\lambda_{\max}}) + \beta_2 (\mathrm{I} - \frac{\mathrm{L}_{\text{norm}}}{\lambda_{\max}})^2 + ... + \beta_H (\mathrm{I} - \frac{\mathrm{L}_{\text{norm}}}{\lambda_{\max}})^H. \tag{14}$$

Considering that L_{norm} is a symmetric real matrix, it can be decomposed as $\mathrm{L}_{\text{norm}} = \mathrm{U}\Lambda\mathrm{U}^\top$, then Eq. (14) can be rewritten as

$$\mathrm{M}_H = \mathrm{U}(\beta_1 (\mathrm{I} - \frac{\Lambda}{\lambda_{\max}}) + \beta_2 (\mathrm{I} - \frac{\Lambda}{\lambda_{\max}})^2 + ... + \beta_H (\mathrm{I} - \frac{\Lambda}{\lambda_{\max}})^H)\mathrm{U}^\top. \tag{15}$$

Here Λ is the diagonal eigenvalue matrix of L_{norm}, and the h-th term $(1 \leq h \leq H)$ in Eq. (15) indicates a frequency response function of $(1 - \frac{\lambda_\mathrm{i}}{\lambda_{\max}})^h$. For any λ_i $(1 \leq i \leq N)$, $\frac{\lambda_i}{\lambda_{\max}} \leq 1$ holds. Therefore, the frequency response of every term in Eq. (15) monotonically decreases w.r.t. λ_i. This indicates that, for each term, when it is multiplied by a graph signal, the higher frequency components of the graph signal are more weakened compared with the lower frequency components. In conclusion, multiplying the propagation matrix M_H with any graph signal equals to the operation of low-pass filtering when $\alpha = \frac{1}{\lambda_{\max}}$ and all $\beta_i > 0$. □

Table 1: The statistics and basic information about the three real-world datasets adopted for experimental evaluation.

Dataset	German Credit	Recidivism	Credit Defaulter
# Nodes	1,000	18,876	30,000
# Edges	22,242	321,308	1,436,858
# Attributes	27	18	13
Avg. degree	44.48	34.04	95.79
Sens.	Gender	Race	Age
Label	good / bad	bail / no bail	default / no default

Based on Theorem 1, we propose the corollary below to build connections between attribute bias and structural bias.

Corollary 1. *The attribute bias contained in the low frequency components of an attributed network is equivalent to structural bias.*

Considering that structural bias is defined based on the low-pass filtered attribute bias (according to Theorem 1), the corollary is self-evident. At the same time, considering that the frequencies and the corresponding basis of a network data changes when A is optimized to be $\tilde{\mathrm{A}}$ [10], the basic goal of EDITS can also be interpreted as: *debiasing the full spectrum of a graph signal, and learning better frequencies together with the corresponding basis to further mitigate the bias existed in the lower frequency components of the graph signal.*

6 EXPERIMENTAL EVALUATIONS

We conduct experiments on both real-world and synthetic datasets to evaluate the effectiveness of EDITS. In particular, we answer the following research questions. **RQ1:** How well can EDITS mitigate the bias in attributed networks together with the outcome of different GNN variants for the downstream task? **RQ2:** How well can EDITS balance utility maximization and bias mitigation compared with other debiasing baselines tailored for a specific GNN?

6.1 Downstream Task and Datasets

Downstream Task. Here we choose the widely adopted *node classification* task to assess the effectiveness of our proposed framework. **Datasets.** We use two types of datasets in our experiments, including three real-world datasets and two synthetic datasets. Statistics of the real-world datasets are shown in Table 1. We present the details of the two kinds of datasets as follows.

- **Real-world Datasets.** We include three real-world datasets, namely German Credit network, Recidivism network, and Credit Defaulter network [2]. Specifically, in *German Credit*, nodes represent clients in a German bank, and edges are formed between clients if their credit accounts are similar. With "gender" being the sensitive attribute, the node classification task here aims to classify the credit risk of the clients as high or low. In *Recidivism*, nodes are defendants released on bail during 1990-2009. Nodes are connected based on the similarity of past criminal records and demographics. The task is to classify defendants into bail vs. no bail with "race" (black and white) being the sensitive attribute. In the *Credit Defaulter*, nodes are credit card users and they are connected based on the pattern similarity of their purchases and payments. Here "age" is the sensitive attribute, and the task is to predict whether a user will default on credit card payment.

- **Synthetic Datasets.** To carry out ablation study on the effectiveness of each module in EDITS, we also use two datasets created similarly as those in Section 2. One network has biased attributes and unbiased structure, while the other network is on the opposite. We add eight extra attribute dimensions besides the two attribute dimensions for both datasets. The attribute value in the extra attribute dimensions are generated uniformly between 0 and 1. For label generation, we compute the sum of the first two extra attribute dimensions. Then, we add noises to the sum values following $\mathcal{N}(0, 0.1^2)$, and rank them by the values in a descending order. The labels of the top-ranked 50% individuals are set as 1, while the labels of the other 50% are set as 0. The goal of the downstream task is to predict the labels.

6.2 Experimental settings

GNN Models. To investigate whether the debiased result of EDITS can make outcomes of different GNNs less biased, we adopt three popular GNN variants to make predictions for the downstream task in our experiments: GCN [29], GraphSAGE [20] and GIN [60].

Baselines. Since there is no existing work directly debiasing attributed network in a model-agnostic way, here we choose two state-of-the-art GNN-based debiasing approaches for comparison, namely FairGNN [12] and NIFTY [2]. (1) *FairGNN.* It is a debiasing method based on adversarial training. A discriminator is trained to distinguish the representations between demographic groups. The goal of FairGNN is to train a GNN that fools the discriminator for bias mitigation; (2) *NIFTY.* It is a recently proposed GNN-based debiasing framework. With counterfactual perturbation on the sensitive attribute, bias is mitigated via learning node representations that are invariant to the sensitive attribute. It should be noted that both of them take GNNs in the downstream task as their backbones. While on the other hand, EDITS attempts at debiasing attributed network without referring to the output of downstream GNN models (i.e., EDITS is model-agnostic). The hyper-parameters of EDITS are tuned only based on our proposed bias metrics applied on attributed networks. Therefore, our studied problem is much more difficult than the model-oriented debiasing of the two baselines.

Evaluation Metrics. We evaluate model performance from two perspectives: model utility and bias mitigation. A good performance means low bias and high model utility. We introduce the adopted metrics for model utility and bias mitigation: (1) *Model Utility Metrics.* For node classification, we use the area under receiver operating characteristic curve (AUC) and F1 score as the indicator of model utility; (2) *Bias Mitigation Metrics.* We use two widely-adopted metrics to show to what extent the bias in the output of different GNNs are mitigated [5, 12, 38]. For both metrics, a lower value means better bias mitigation performance. The metric measuring bias from the perspective of *Statistical Parity* is given by

$$\Delta_{SP} = |P(\hat{y} = 1|s = 0) - P(\hat{y} = 1|s = 1)|. \tag{16}$$

The metric from the perspective of *Equality of Opportunity* is

$$\Delta_{EO} = |P(\hat{y} = 1|y = 1, s = 0) - P(\hat{y} = 1|y = 1, s = 1)|. \tag{17}$$

Table 2: Attribute and structural bias comparison between original networks and debiased ones from EDITS. Lower value for both metrics indicates better bias mitigation. All values are in scale of $\times 10^{-3}$. Best ones are marked in bold.

	Attribute Bias		Structural Bias	
	Vanilla	EDITS	Vanilla	EDITS
German	6.33	**2.38** (−62.4%)	10.4	**3.54** (−66.0%)
Credit	2.46	**0.56** (−77.2%)	4.45	**2.36** (−47.0%)
Recidivism	0.95	**0.39** (−58.9%)	1.10	**0.52** (−52.7%)

6.3 Implementation Details

EDITS is implemented using Pytorch [44] and optimized via RMSprop optimizer [23]. For GNN models (i.e., GCN, GraphSAGE and GIN) and baseline methods (i.e., FairGNN and NIFTY) in our experiments, we use their released Pytorch implementations for a fair comparison. In the training of EDITS, we set the training epochs as 100 for Recidivism and 500 for other datasets. The learning rate is set as 3×10^{-3} for epochs under 400 and 1×10^{-3} for those above. α is set as 0.5 considering that $\lambda_{\max} = 2$ [10]. For a more stable optimization, the three modules in EDITS are updated alternatively. The attributed networks output by EDITS are used for GNN training to evaluate the performance of EDITS, while the performance of the Vanilla method is obtained by feeding GNNs with the original networks. To train GNNs, we fix the training epochs to be 1,000, with the learning rate of 1×10^{-3}. Dropout rate and hidden channel number is set as 0.05 and 16, respectively. All GNNs and baseline methods are optimized with Adam optimizer [28].

6.4 Debiasing Attributed Network for GNNs

To answer **RQ1**, we first evaluate the effectiveness of EDITS on reducing the bias measured by the two proposed metrics and traditional bias metrics with different GNN backbones. The attribute and structural bias of the three real-world datasets before and after being debiased by EDITS are shown in Table 2. The quantitative comparison on Δ_{SP} and Δ_{EO} between GNNs trained on debiased networks from EDITS and original networks is also presented in Table 3. We make the following observations from the two Tables:

- From the perspective of bias mitigation in the attributed network (i.e., attribute bias and structural bias), EDITS exhibits significant advantages over the Vanilla approach as indicated by Table 2. This verifies the effectiveness of EDITS on reducing the bias existing in the attributed network data.
- From the perspective of bias mitigation in the downstream task (i.e., Δ_{SP} and Δ_{EO}), we observe from Table 3 that EDITS achieves desirable bias mitigation performance with little utility sacrifice in all cases compared with GNNs with original network as input (i.e., the vanilla one). This verifies that attributed networks debiased by EDITS can generally mitigate the bias in the outcome of different GNNs.

Table 3: Performance comparison on model utility and bias mitigation between GNNs with original network input (denoted as Vanilla) and those with debiased network input from EDITS (denoted as EDITS) on three real-world datasets. ↑ denotes the larger, the better; ↓ denotes the opposite. Best results are marked in bold.

		GCN		GraphSAGE		GIN	
		Vanilla	EDITS	Vanilla	EDITS	Vanilla	EDITS
German	**AUC ↑**	**74.46 ± 0.7%**	71.01 ± 1.3%	**75.28 ± 2.1%**	73.21 ± 0.5%	71.35 ± 1.7%	**71.51 ± 0.6%**
	F1 ↑	81.54 ± 0.9%	**82.43 ± 0.69%**	**81.52 ± 1.0%**	80.62 ± 1.5%	83.08 ± 0.9%	**83.78 ± 0.4%**
	Δ_{SP} ↓	43.14 ± 2.5%	**2.04 ± 1.3%**	26.83 ± 0.5%	**8.30 ± 3.1%**	18.55 ± 2.0%	**1.26 ± 0.7%**
	Δ_{EO} ↓	33.75 ± 0.4%	**0.63 ± 0.39%**	20.66 ± 3.0%	**3.75 ± 3.3%**	11.27 ± 3.5%	**2.87 ± 1.4%**
Credit	**AUC ↑**	**73.62 ± 0.3%**	70.16 ± 0.6%	74.99 ± 0.2%	**75.28 ± 0.5%**	**73.82 ± 0.4%**	72.06 ± 0.9%
	F1 ↑	**81.86 ± 0.1%**	81.44 ± 0.2%	82.31 ± 0.7%	**83.39 ± 0.3%**	82.11 ± 0.1%	**85.10 ± 0.7%**
	Δ_{SP} ↓	12.93 ± 0.1%	**9.13 ± 1.2%**	17.03 ± 3.3%	**12.25 ± 0.2%**	12.18 ± 0.3%	**8.79 ± 5.6%**
	Δ_{EO} ↓	10.65 ± 0.0%	**7.88 ± 1.0%**	15.31 ± 4.0%	**9.58 ± 0.1%**	9.48 ± 0.3%	**7.19 ± 3.8%**
Recidivism	**AUC ↑**	**86.91 ± 0.4%**	85.96 ± 0.3%	88.12 ± 1.4%	**88.15 ± 0.9%**	**82.40 ± 0.8%**	81.55 ± 1.5%
	F1 ↑	**78.30 ± 1.0%**	75.80 ± 0.5%	76.23 ± 2.8%	**76.30 ± 1.4%**	70.36 ± 1.9%	**71.09 ± 2.3%**
	Δ_{SP} ↓	7.89 ± 0.3%	**5.39 ± 0.2%**	2.42 ± 1.2%	**0.79 ± 0.5%**	9.97 ± 0.7%	**4.98 ± 0.9%**
	Δ_{EO} ↓	5.58 ± 0.2%	**3.36 ± 0.3%**	2.98 ± 2.2%	**1.01 ± 0.5%**	6.10 ± 1.2%	**5.47 ± 0.7%**

- When comparing bias mitigation performance indicated by Table 2 and Table 3, we can find that the bias in the outcome of GNNs is also mitigated after EDITS mitigates attribute bias and structural bias in the attributed networks. Such consistency verifies the validity of our proposed metrics on measuring the bias existed in the attributed network data.

6.5 Comparison with Other Debiasing Models

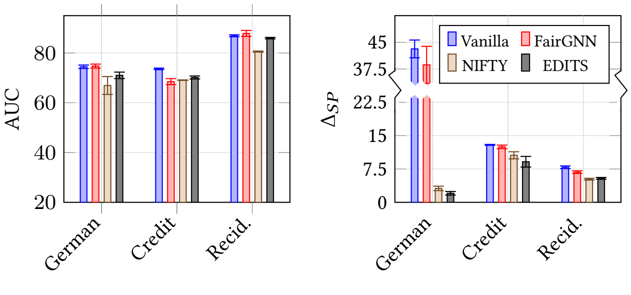
To answer **RQ2**, we then compare the balance between model utility and bias mitigation with other baselines on a given GNN. Here we present the comparison of AUC and Δ_{SP} based on GCN in Fig. (3). Similar results can be obtained for other GNNs, which are omitted due to space limit. The following observations can be made:

- From the perspective of model utility (indicated by Fig. (3a)), EDITS and the baselines methods achieve comparable results with the vanilla GCN. This implies that the debiasing process of EDITS preserves as much useful information for the downstream task as the original attributed network.
- From the perspective of bias mitigation (indicated by Fig. (3b)), all baselines achieve effective bias mitigation. For debiasing in the downstream task, NIFTY performs better than FairGNN. Compared with debiasing in the downstream task, debiasing the attributed network is more difficult due to the lack of supervision signals from the downstream task. Interestingly, EDITS performs better than both two baselines on bias mitigation. This verifies the superior performance of EDITS on debiasing the attributed network data.
- From the perspective of balancing the model utility and bias mitigation, EDITS achieves comparable model utility with vanilla GNNs and other baselines, but exhibits better bias mitigation performance. Consequently, we argue that EDITS achieves superior performance on balancing the model utility and bias mitigation compared with other baselines.

6.6 Ablation Study

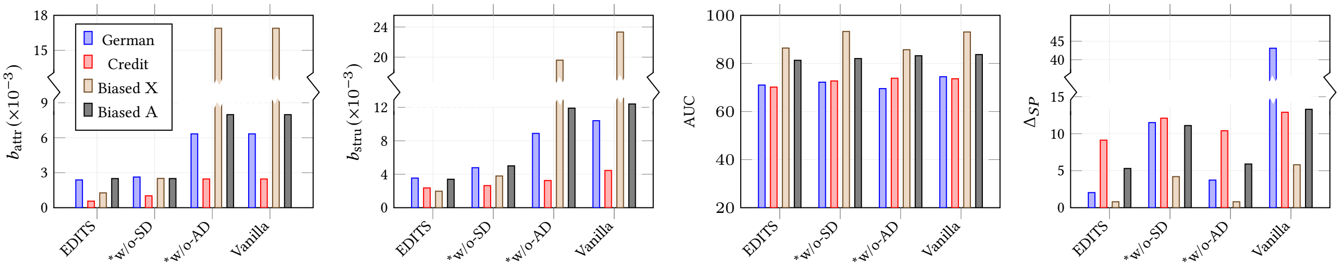
To evaluate the effectiveness of the two debiasing modules (i.e., attribute debiasing module and structural debiasing module) in EDITS, here we investigate how each of them individually contributes to bias mitigation under our proposed bias metrics and the bias metrics in the downstream task. We choose GCN as the GNN model in our downstream task. For better visualization purpose, the two datasets showing large attribute bias and structural bias (i.e., *German* and *Credit*) are selected for experiments. Besides, to better demonstrate the functionality of the two debiasing modules, we also adopt the two synthetic datasets we mentioned in Section 2 (i.e., the networks with either only biased attributes or only biased structure), which are further modified according to Section 6.1. Based on the four selected datasets, four different variants of EDITS are tested, namely EDITS with both debiasing modules, EDITS without the structural debiasing module (i.e., *w/o-SD), EDITS without the attribute debiasing module (i.e., *w/o-AD), vanilla GCN model without debiased input (i.e., Vanilla). We present their performance of attribute bias, structural bias, AUC and Δ_{SP} on the four datasets in Fig. (4). We make the following observations:

- The value of *attribute bias* can be reduced with attribute debiasing module of EDITS, which maintains the model utility (i.e., AUC) but reduces Δ_{SP} in the downstream task.
- The value of *structural bias* can be reduced with both attribute debiasing and structural debiasing module. With only structural debiasing, EDITS still maintains comparable model utility but reduces Δ_{SP} in the downstream task.



(a) Comparison on AUC **(b) Comparison on Δ_{SP}**

Figure 3: Performance comparison between EDITS and baselines on utility (AUC) and bias mitigation (Δ_{SP}) on GCN.



(a) Attribute bias **(b) Structural bias** **(c) Model utility performance** **(d) Fairness performance**

Figure 4: Performance EDITS and its variants on two real-world datasets and two synthetic datasets. EDITS denotes that both debiasing modules are included; *w/o-SD means EDITS without sturctural debiasing module; *w/o-AD means EDITS is without attribute debiasing module; Vanilla means applying GNN with the original attributed network as input.

- Although both attribute debiasing and structural debiasing module help mitigate *structural bias*, only debiasing the network structure achieves better bias mitigation performance on all four datasets compared with only debiasing the attributes as implied by Fig. (4d). This demonstrates the indispensability of the structural debiasing module in EDITS.

7 RELATED WORK

Mitigating Bias in Machine Learning. Bias can be defined differently in machine learning [4, 14, 22, 35, 39, 58]. Commonly used bias notions can be broadly categorized into *group fairness* and *individual fairness* [16]. Group fairness emphasizes that algorithms should not yield discriminatory outcomes for any specific demographic group [16]. Such group are usually determined by attributes that are reluctant or illegal to be shared (e.g., gender or race) [26], i.e., sensitive attributes. Existing debiasing approaches work in any of the three data flow stages, i.e., pre-processing, processing and post-processing stage. In pre-processing stage, a common method is to re-weight training samples from different groups to mitigate bias before model training [26]. Perturbing data distributions between groups is another popular approach to debias the data in the pre-processing stage [57]. In processing stage, a popular method is to add regularization terms to disentangle the outcome from sensitive attribute [37, 50] or minimize the outcome difference between groups [1]. Besides, utilizing adversarial learning to remove sensitive information from representations is also widely adopted [18]. In post-processing stage, bias in outcomes is usually mitigated by constraining the outcome to follow a less biased distribution [22, 45, 63]. Usually, all above-mentioned approaches are evaluated via measuring how much certain fairness notion is violated, e.g., *Statistical Parity* [16], *Equality of Opportunity*, *Equality of Odds* [21] and *Counterfactual Fairness* [31]. Different from group fairness, individual fairness focuses on treating similar individuals similarly [16, 61]. The similarity can be given by oracle similarity scores from specialists [32]. Most existing debiasing methods based on individual fairness work in the processing stage. For example, constraint can enforce similar predictions between similar instances [25, 32], and *Consistency* is a popular evaluation metric [32, 33].

Mitigating Bias in Graph Mining. Efforts have been made to mitigate bias in graph mining algorithms, where these works can be broadly categorized into either focusing on *group fairness* or *individual fairness*. For group fairness, adversarial learning can be adopted to learn less biased representations that fool the discriminator [6, 7, 11]. Rebalancing between groups is also a popular approach to mitigate bias [8, 19, 36, 46, 53]. For example, Rahman et al. mitigate bias via rebalancing the appearance rate of minority groups in random walks [48]. Projecting the embeddings onto a hyperplane orthogonal to the hyperplane of sensitive attributes is another approach for bias mitigation based on group fairness. Compared with other methods, it theoretically guarantees that embeddings are disentangled from sensitive attributes [42]. Compared with mitigating bias focusing on group fairness, only few work promotes individual fairness in graphs. To the best of our knowledge, Kang et al. first propose to systematically debias multiple graph mining algorithms based on individual fairness [27]. Different from all previous debiasing approaches, this paper proposes to directly debias the attributed network data in a model-agnostic way based on the analysis of information propagation mechanism in GNNs.

8 CONCLUSION

GNNs are playing increasingly important roles in many practical scenarios. Nevertheless, there is an increasing societal concern that GNNs could yield discriminatory decisions towards certain demographic groups. Such discrimination can be harmful and even illegal under certain circumstances. Existing debiasing approaches are mainly tailored for a specific GNN. Adapting these methods to different GNNs can be costly, as they need to be trained and fine-tuned for different GNNs. Different from them, in this paper, we propose to directly debias the attributed network data for GNNs. With preliminary analysis on the source of bias and interaction between bias in different data modalities, we formally define two kinds of bias together with their corresponding metrics, and formulate a novel problem of debiasing attributed network for GNNs. To tackle this problem, we then propose a principled framework named EDITS for model-agnostic debiasing. Experimental results demonstrate that EDITS exhibits superior performance on mitigating the bias and maintaining the model utility in the downstream task compared with other state-of-the-art approaches.

9 ACKNOWLEDGEMENTS

This material is, in part, supported by the National Science Foundation (NSF) under grants #2006844.

REFERENCES

[1] Alekh Agarwal, Alina Beygelzimer, Miroslav Dudík, John Langford, and Hanna Wallach. 2018. A reductions approach to fair classification. In *International Conference on Machine Learning*. PMLR, 60–69.

[2] Chirag Agarwal, Himabindu Lakkaraju, and Marinka Zitnik. 2021. Towards a Unified Framework for Fair and Stable Graph Representation Learning. *arXiv preprint arXiv:2102.13186* (2021).

[3] Martín Arjovsky, Soumith Chintala, and Léon Bottou. 2017. Wasserstein GAN. *CoRR* abs/1701.07875 (2017). arXiv:1701.07875 http://arxiv.org/abs/1701.07875

[4] Solon Barocas, Moritz Hardt, and Arvind Narayanan. 2017. Fairness in machine learning. *NeurIPS Tutorial* 1 (2017).

[5] Alex Beutel, Jilin Chen, Zhe Zhao, and Ed H Chi. 2017. Data decisions and theoretical implications when adversarially learning fair representations. *arXiv preprint arXiv:1707.00075* (2017).

[6] Avishek Bose and William Hamilton. 2019. Compositional Fairness Constraints for Graph Embeddings. In *Proceedings of the 36th ICML (ICML '19)*.

[7] Avishek Joey Bose and William L. Hamilton. 2019. Compositional Fairness Constraints for Graph Embeddings. In *Proceedings of the 36th ICML (ICML '19)*.

[8] R. Burke, N. Sonboli, M. Mansoury, and A. Ordoñez-Gauger. 2017. Balanced neighborhoods for fairness-aware collaborative recommendation. (2017).

[9] Toon Calders, Faisal Kamiran, and Mykola Pechenizkiy. 2009. Building classifiers with independency constraints. In *2009 IEEE International Conference on Data Mining Workshops*. IEEE, 13–18.

[10] Fan RK Chung and Fan Chung Graham. 1997. *Spectral graph theory*. Number 92. American Mathematical Soc.

[11] Enyan Dai and Suhang Wang. 2020. FairGNN: Eliminating the Discrimination in Graph Neural Networks with Limited Sensitive Attribute Information. *arXiv preprint arXiv:2009.01454* (2020).

[12] Enyan Dai and Suhang Wang. 2020. Say No to the Discrimination: Learning Fair Graph Neural Networks with Limited Sensitive Attribute Information. arXiv:2009.01454 [cs.LG]

[13] Kien Do, Truyen Tran, and Svetha Venkatesh. 2019. Graph transformation policy network for chemical reaction prediction. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 750–760.

[14] Mengnan Du, Fan Yang, Na Zou, and Xia Hu. 2019. Fairness in deep learning: A computational perspective. *arXiv preprint arXiv:1908.08843* (2019).

[15] C. Dwork, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Richard Zemel. 2012. Fairness through awareness. In *Proceedings of the 3rd ITCS (ITCS '12)*.

[16] Cynthia Dwork, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Richard S. Zemel. 2012. Fairness through awareness. In *Innovations in Theoretical Computer Science 2012, Cambridge, MA, USA, January 8-10, 2012*.

[17] Michael D Ekstrand and Daniel Kluver. 2021. Exploring author gender in book rating and recommendation. *User Modeling and User-Adapted Interaction* (2021).

[18] Yanai Elazar and Yoav Goldberg. 2018. Adversarial removal of demographic attributes from text data. *arXiv preprint arXiv:1808.06640* (2018).

[19] Golnoosh F., Pigi K., Spencer K. T., Sriram S., and Lise G. 2018. A Fairness-aware Hybrid Recommender System. *CoRR* abs/1809.09030 (2018).

[20] Will Hamilton, Zhitao Ying, and Jure Leskovec. 2017. Inductive representation learning on large graphs. In *Proceedings of the 31st NeurIPS (NIPS '17)*.

[21] Moritz Hardt, Eric Price, and Nati Srebro. 2016. Equality of Opportunity in Supervised Learning. In *Proceedings of the 30th NeurIPS (NeurIPS '16)*.

[22] Moritz Hardt, Eric Price, and Nati Srebro. 2016. Equality of opportunity in supervised learning. In *Proceedings of the 30th NeurIPS (NIPS '16)*.

[23] G. Hinton, N. Srivastava, and K. Swersky. 2012. Neural networks for machine learning lecture 6a overview of mini-batch gradient descent. *Cited on* 14, 8 (2012).

[24] Zeinab S Jalali, Weixiang Wang, Myunghwan Kim, Hema Raghavan, and Sucheta Soundarajan. 2020. On the Information Unfairness of Social Networks. In *SIAM International Conference on Data Mining*. SIAM, 613–521.

[25] Christopher Jung, Michael Kearns, Seth Neel, Aaron Roth, Logan Stapleton, and Zhiwei Steven Wu. 2019. Eliciting and enforcing subjective individual fairness. *arXiv preprint arXiv:1905.10660* (2019).

[26] F. Kamiran and T. Calders. 2012. Data preprocessing techniques for classification without discrimination. *Knowledge and Information Systems* 33, 1 (2012), 1–33.

[27] J. Kang, J. He, R. Maciejewski, and H. Tong. 2020. InFoRM: Individual Fairness on Graph Mining. In *Proceedings of the 26th SIGKDD (KDD '20)*.

[28] Diederik P. Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In *Proceedings of the 3rd ICLR (ICLR '15)*.

[29] Thomas N Kipf and Max Welling. 2016. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907* (2016).

[30] Thomas N Kipf and Max Welling. 2016. Variational graph auto-encoders. *arXiv preprint arXiv:1611.07308* (2016).

[31] Matt J Kusner, Joshua Loftus, Chris Russell, and Ricardo Silva. 2017. Counterfactual fairness. In *Advances in Neural Information Processing Systems*. 4066–4076.

[32] Preethi L., K. P. Gummadi, and G. Weikum. 2019. Operationalizing Individual Fairness with Pairwise Fair Representations. *Proc. VLDB Endow.* (2019).

[33] Preethi Lahoti, Krishna P Gummadi, and Gerhard Weikum. 2019. ifair: Learning individually fair data representations for algorithmic decision making. In *Proceedings of the 35th ICDE (ICDE '19)*. IEEE.

[34] Jure Leskovec and Julian J Mcauley. 2012. Learning to discover social circles in ego networks. In *Proceedings of the 26th NeurIPS (NIPS '12)*.

[35] Peizhao Li, Yifei Wang, Han Zhao, Pengyu Hong, and Hongfu Liu. 2020. On dyadic fairness: Exploring and mitigating bias in graph connections. In *International Conference on Learning Representations*.

[36] Peizhao Li, Yifei Wang, Han Zhao, Pengyu Hong, and Hongfu Liu. 2021. On dyadic fairness: Exploring and mitigating bias in graph connections. In *Proceedings of International Conference on Learning Representations*.

[37] Frederick Liu and Besim Avci. 2019. Incorporating priors with feature attribution on text classification. *arXiv preprint arXiv:1906.08286* (2019).

[38] Christos Louizos, Kevin Swersky, Yujia Li, Max Welling, and Richard Zemel. 2015. The variational fair autoencoder. *arXiv preprint arXiv:1511.00830* (2015).

[39] Ninareh M., Fred M., Nripsuta S., Kristina L., and Aram G. 2019. A survey on bias and fairness in machine learning. *arXiv preprint arXiv:1908.09635* (2019).

[40] Farzan Masrour, Tyler Wilson, Heng Yan, Pang-Ning Tan, and Abdol Esfahanian. 2020. Bursting the Filter Bubble: Fairness-Aware Network Link Prediction. In *AAAI Conference on Artificial Intelligence*. 841–848.

[41] Hoang NT and Takanori Maehara. 2019. Revisiting graph neural networks: All we have is low-pass filters. *arXiv preprint arXiv:1905.09550* (2019).

[42] John Palowitch and Bryan Perozzi. 2019. MONET: Debiasing Graph Embeddings via the Metadata-Orthogonal Training Unit. *CoRR* abs/1909.11793 (2019).

[43] Eli Pariser. 2011. *The filter bubble: How the new personalized web is changing what we read and how we think*. Penguin.

[44] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. 2017. Automatic differentiation in pytorch. (2017).

[45] Geoff Pleiss, Manish Raghavan, Felix Wu, Jon Kleinberg, and Kilian Q Weinberger. 2017. On fairness and calibration. *arXiv preprint arXiv:1709.02012* (2017).

[46] Tahleen A. R., Bartlomiej S., Michael B., and Yang Zhang. 2019. Fairwalk: Towards Fair Graph Embedding. In *Proceedings of the 28th IJCAI (IJCAI '19)*.

[47] Lawrence R Rabiner, Bernard Gold, and CK Yuen. 2016. *Theory and application of digital signal processing*. Prentice-Hall.

[48] T. A Rahman, B. Surma, M. Backes, and Y. Zhang. 2019. Fairwalk: Towards Fair Graph Embedding. In *Proceedings of the 28th IJCAI (IJCAI '19)*.

[49] L. Ribeiro, P. Saverese, and D. Figueiredo. 2017. struc2vec: Learning node representations from structural identity. In *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*. 385–394.

[50] Andrew Slavin Ross, Michael C Hughes, and Finale Doshi-Velez. 2017. Right for the right reasons: Training differentiable models by constraining their explanations. *arXiv preprint arXiv:1703.03717* (2017).

[51] V. Shumovskaia, K. Fedyanin, I. Sukharev, D. Berestnev, and M. Panov. 2021. Linking bank clients using graph neural networks powered by rich transactional data. *International Journal of Data Science and Analytics* (2021), 1–11.

[52] Lei Tang and Huan Liu. 2009. Relational learning via latent social dimensions. In *Proceedings of the 15th SIGKDD (KDD '09)*.

[53] X. Tang, H. Yao, Y. Sun, Y. Wang, J. Tang, C. Aggarwal, P. Mitra, and S. Wang. 2020. Investigating and Mitigating Degree-Related Biases in Graph Convoltuional Networks. In *Proceedings of the 29th CIKM (CIKM '20)*.

[54] Petar V., Guillem C., Arantxa C., Adriana R., Pietro L., and Yoshua B. 2017. Graph attention networks. *arXiv preprint arXiv:1710.10903* (2017).

[55] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. 2017. Graph attention networks. *arXiv preprint arXiv:1710.10903* (2017).

[56] Cédric Villani. 2008. *Optimal transport: old and new*. Vol. 338. Springer Science & Business Media.

[57] Hao Wang, Berk Ustun, and Flavio Calmon. 2019. Repairing without retraining: Avoiding disparate impact with counterfactual distributions. In *International Conference on Machine Learning*. PMLR, 6618–6627.

[58] Y. Wu, L. Zhang, and X. Wu. 2019. Counterfactual Fairness: Unidentification, Bound and Algorithm. In *International Joint Conference on Artificial Intelligence*.

[59] Bingbing Xu, Huawei Shen, Bingjie Sun, Rong An, Qi Cao, and Xueqi Cheng. 2021. Towards Consumer Loan Fraud Detection: Graph Neural Networks with Role-Constrained Conditional Random Field. (2021).

[60] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. 2018. How powerful are graph neural networks? *arXiv preprint arXiv:1810.00826* (2018).

[61] Rich Zemel, Yu Wu, Kevin Swersky, Toni Pitassi, and Cynthia Dwork. 2013. Learning fair representations. In *Proceedings of the 30th ICML (ICML '13)*.

[62] Muhan Zhang and Yixin Chen. 2018. Link Prediction Based on Graph Neural Networks. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, Samy Bengio, Hanna M. Wallach, Hugo Larochelle, Kristen Grauman, Nicolò Cesa-Bianchi, and Roman Garnett (Eds.). 5171–5181. https://proceedings.neurips.cc/paper/2018/hash/53f0d7c537d99b3824f0f99d62ea2428-Abstract.html

[63] Jieyu Zhao, Tianlu Wang, Mark Yatskar, Vicente Ordonez, and Kai-Wei Chang. 2017. Men also like shopping: Reducing gender bias amplification using corpus-level constraints. *arXiv preprint arXiv:1707.09457* (2017).